#13

49658-0034

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:	John WAINWRIGHT		
Serial No.:	09/426,143	Examiner	Chante E. HARRISON
Filing Date:	October 22, 1999	Art Unit:	2672

For: SPECIFYING OPERATIONS TO BE APPLIED TO THE ATTRIBUTES OF A SET OF OBJECTS

Commissioner for Patents
Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Sir:

This Appeal Brief is submitted in support of the Notice of Appeal filed on July 15, 2003.

I. REAL PARTY IN INTEREST

Autodesk, Inc. is the real party in interest.

II. RELATED APPEALS AND INTERFERENCES

To the present knowledge of Appellant and Appellant's legal representative, there are currently no related appeals or interference proceedings in progress which will directly affect, or be directly affected by, or have a bearing on the Board's decision in the present Appeal.

III. STATUS OF CLAIMS

Claims 1-20 are pending in this Application. Claims 1-5, 7-9, 12-16, 18 and 19 stand rejected under 35 U.S.C. §102(e) as allegedly being anticipated by Perlin (U.S. Patent No.

6,285,380). Claims 6, 10, 11, 17 and 20 stand objected to as allegedly being dependent upon a rejected base claim. Claims 1-5, 7-9, 12-16, 18 and 19 are the subject of this appeal.

IV. STATUS OF AMENDMENTS

No amendments were filed after the final Office Action. An examiner interview was held on July 10, 2003, at which no claim amendments were proposed.

V. SUMMARY OF THE INVENTION

BACKGROUND CONTEXT

To assist in understanding the invention, some background context is first provided.

To generate animations, a developer creates computer models of the entities using computer aided design systems (CAD). These models are used to emulate the movement, color, and shape of animated entities. Models are often composed of graphical components that represent the shapes and surfaces that make up modeled entities. A graphical component is a set of data, procedures, or a combination thereof, used to represent a geometry, such as a curve or the surface of a car. A graphical component may consist of multiple other graphical components, to represent more complex geometry, such as a car or human individual.

Models are often built by users using a Computer Aid Design ("CAD") system. Typically, a user enters commands and data through a graphical user interface ("GUI"), which are used to create frames for animations. A frame is the state of a set of graphical components at a particular point in time. Animations are generated by displaying a sequence of frames at a particular frequency. For example, a sequence of frames may be used to animate marbles rolling across the room from the left side of the room to the right. Each frame of the sequence would include a graphical component for each of the marbles. In the first frame the marbles are at the far left of the room. In the second frame, the marbles are positioned a little closer to the right, and so forth. The marbles are rendered in positions that are shifted in each of the frames, which are displayed in rapid sequence to animate the marbles rolling across the room.

To generate a sequence of frames, a user, through the GUI interface on a CAD system, may generate data specifying the state of the frames. For a given frame, the user moves each of the marbles to a different position by dragging an image of the marble displayed in the GUI, storing data for the frame, then dragging each of the marbles to their next position, and storing data for another frame, and repeating these manipulations for each of the remaining frames. Consequently, to generate a sequence of frames, a user may repetitively perform the same GUI manipulations.

Often, user input that is created by repetitively performing the same kinds of manipulations may be entered more efficiently through the use of a scripting language. Very often developers of scripts program tasks that apply the same operations to each graphical component in a set of graphical components. A task that involves performing the same or substantially similar operation on a set of graphical components is referred to herein as a duplicated task. A duplicated task may be programmed by explicitly writing lines that each specify the same operation, each line referencing a particular graphical component in the set.

Obviously, writing very similar lines of code that each specify an identical operation is a repetitive task. In addition, writing a group of such lines to program a duplicated task may frustrate another purpose for writing scripts or programs in a computer language, which is to describe a task or algorithm to a reader of the script or program. Multiple lines of code that specify substantially the same task describe a duplicated task inconcisely, and thus impede comprehension of the script.

SUMMARY OF EMBODIMENTS

Embodiments of the present invention provide, generally, techniques for programming duplicated tasks that do not require programming constructs, such as a "for" loop, for repeatedly executing the same operation. More specifically, a statement (e.g., statement 200, 220, 240, 260) (page 9, lines 17 and 18) specifies an operation (page 10, line 8 through page 11, line 2), which is identified by an operation identifier (e.g., 212, 222, 242, 252) (page 10, line 7 through page 11, line 2), to perform on an attribute (page 10, lines 10-17), which is identified by an attribute

identifier (e.g., 216) (page 9, lines 4-8), of a set of objects (page 8, line 21 through page 9, line 2).

The statement may specify the set via (A) pattern matching criteria (e.g., 214, 224) for identifiers associated with objects (specification page 12, line 17 through page 15, line 23; FIG. 2), or (B) an identifier (e.g., 244, 254) associated with a collection of objects (specification page 16, line 1 through page 17, line 21; FIG. 2). Objects that are associated with identifiers that satisfy the pattern matching criteria or with the collection of objects, respectively, are identified. For each identified object, the operation is applied to the attribute of the object.

In other words, in one embodiment, with execution of a single statement,

(1) objects can be "filtered" based on matching object identifiers with pattern matching criteria, and

(2) the operation specified in the statement is performed on the attribute identified in the statement, for each object of the set of objects that is identified as satisfying the pattern matching criteria via the "filtering".

VI. ISSUES

(A) Are Claims 1-5, 7-9, 12-16, 18 and 19 anticipated under 35 U.S.C. §102(e) by Perlin et al., U.S. Patent No. 6,285,380 ("*Perlin*")?

VII. GROUPING OF CLAIMS

It is submitted that Claims 1-5, 7-9, 12-16, 18 and 19 do not fall or stand together and the following groupings are asserted:

GROUP 1: Claims 1, 4, 5, 12, 15 and 16;

GROUP 2: Claims 2, 3, 13 and 14;

GROUP 3: Claim 7;

GROUP 4: Claims 8 and 18; and

GROUP 5: Claims 9 and 19.

VIII. ARGUMENTS

A. THE *PERLIN* PATENT IS AN IMPROPER REFERENCE UNDER 35 U.S.C. §102(e)

It is well established, and 35 U.S.C. §102(e) plainly states, *inter alia*, that "a person shall be entitled to a patent unless the invention was described in ... a patent granted on an application for patent by another filed in the United States before the invention by the applicant for the patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for the purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language" (emphasis added).

The *Perlin* patent appears to have developed from an international application filed under the referenced treaty (Patent Cooperation Treaty). The PCT international application was filed on August 1, 1997, with a §371 date and **§102(e) date of November 4, 1998**, as indicated on the cover thereof.

The present application claims the benefit of priority to U.S. Provisional Application No. 60/105,512 entitled "3-D Modeling Through A Scripting Language," filed on **October 23, 1998** under 35 U.S.C. § 111(b). The subject matter disclosed in the aforementioned provisional patent application is disclosed in a manner provided by 35 U.S.C. § 112 and, therefore, fully supports each of the features of each of Claims 1-20. Furthermore, the present application includes a specific reference to the aforementioned provisional application, pursuant to 35 U.S.C. § 119(e). The present application was filed less than 12 months after the date on which the aforementioned provisional application was filed. Therefore, the right of priority to U.S. Provisional Patent Application No. 60/105,512 is hereby asserted.

The *Perlin* patent did not grant on an application that was filed in the United States before the invention by the applicant for the patent, as required by 35 U.S.C. §102(e). In

addition, the exception of 35 U.S.C. §102(e) does not afford the *Perlin* patent a §102(e) date prior to the priority date of the present application.

When this fact was addressed during prosecution, the responding Office Action contended that "the prior art relied upon is a continuation in part of patent 6,115,053, which has a priority date of 8/1994". The standard of §102(e) is clearly based on the filing date of a patent application, not its priority date. In addition, for subject matter in a continuation-in-part application to be afforded the right of priority to the parent application to which such priority is claimed, the parent application must disclose the subject matter to which the right to priority is claimed.

Therefore, based on the present application's right of priority to the provisional application, it is respectfully submitted that the *Perlin* reference is an improper reference for a rejection of any claims in the present application, under 35 U.S.C. §102(e). Consequently, any rejection of claims based on *Perlin* as prior art under §102(e) cannot stand.

B. THE LIMITATIONS OF CLAIMS 1-5, 7-9, 12-16, 18 and 19 ARE NOT DISCLOSED IN THE *PERLIN* PATENT

Perlin, entitled "Method and System for Scripting Interactive Animated Actors", is directed towards a system for the creation of behavior-based animated actors. The system provides tools to create actors that respond to users and to each other in real-time, with personalities and moods consistent with the author's goals. Two subsystems are described. An Animation Engine enables authors to create motions and smooth transitions between them, and a Behavior Engine enables authors to create rules governing how actors communicate, change, and make decisions. The combined system provides tools for authoring the "minds" and "bodies" of interactive actors over networks in real-time. (Abstract).

Claim 1 of the application recites the following:

A method of executing an operation on a set of objects, the method comprising the computer-implemented steps of:

detecting that a statement contains

an operation identifier that specifies an operation,

pattern matching criteria, and

an attribute identifier that identifies an attribute; and

executing said statement by

identifying all objects associated with identifiers that satisfy said pattern matching criteria, and

performing said operation on said attribute of each of said objects that satisfy said pattern matching criteria.

It is not disputed that *Perlin* may disclose operations on animated actors, or objects. However, Perlin does not teach the method recited in Claim 1 and, therefore, cannot and does not anticipate Claim 1. Furthermore, *Perlin* does not suggest or motivate the limitations of Claim 1 and, therefore, does not make Claim 1 obvious.

First, *Perlin* does not disclose a statement that includes each of (1) an operation identifier; (2) pattern matching criteria; and (3) an attribute identifier. An example of such a statement is statement 200 of FIG. 2, as follows:

box*.position = [0,0,0].

In the foregoing statement, (1) "=" is the exemplary operation identifier; (2)"box*" is the exemplary pattern matching criteria; and (3) "position" is the exemplary attribute identifier.

The Office Actions relied on col. 13, lines 1-10 of *Perlin* for the alleged teaching of each of the statement features, relying additionally on col. 12, lines 14-24 for the attribute identifier. The Office Actions did not specifically indicate what part of the citation allegedly teaches which features. When discussed in the Examiner Interview, it was stated that "Courage" teaches pattern matching criteria, "Courage Level" teaches an attribute identifier, and "to within 0.5" teaches an operation identifier.

Appellant contends that these properties disclosed in *Perlin* are not analogous to and do not anticipate the features contained in the statement detected in Claim 1. Relative to the cited section, *Perlin* describes the actor's "Courage" property (alleged **pattern matching criteria**) being compared with the "Courage Level" property (alleged **attribute identifier**) associated with the scripts "Flight" and "Flee", where if the actor's "Courage" equals the script's "Courage Level", the decision rule will assign a weight of 1 to that choice of "Flight" or "Flee". If the values are not equal, then a weight between 0 and 1 is assigned based on the difference in values, dropping to 0 when the difference is greater than the "within" range (alleged **operation identifier**). Further, as the actor's "Courage" increases or decreases, so will the actor's tendency toward "Fight" or "Flee", respectively.

Notice that the **pattern matching criteria** of Claim 1 is used to identify all objects associated with **identifiers** that satisfy the pattern matching criteria. Thus, in the foregoing example statement, all objects that have identifiers, e.g., names, that begin with the pattern "box" would be identified as matching the criteria "box*" where "*" is a wildcard character indicator. If "Courage" is considered to be pattern matching criteria as alleged, then in order to anticipate Claim 1 *Perlin* would need to teach that actors, as objects, are identified which have names matching "Courage". There is no such teaching in *Perlin*.

The closest that *Perlin* comes to teaching identifying particular actors seems to be at col. 12, lines 37-42; col. 13, line 63 through col. 14, line 14; and the related descriptions. The referenced sections describe choosing an actor from the group of "Steph", "Bob" and "Sarah" based on "who's interesting", where "who's interesting" is defined by DECISION-RULES that include "influence" or scale values relative to factors such as "Charisma", "Intelligence", "Gender", etc. Perhaps such factors are attributes of the actor objects. So at most, *Perlin* may teach identifying objects based on values or Levels associated with attributes. However, Perlin does not teach identification of objects associated with identifiers that satisfy pattern matching criteria.

Furthermore, Claim 1 requires **performing the operation identified in the statement on the attribute identified in the statement for each object identified** as described above. Since *Perlin* does not teach such identification of objects, it cannot and does not teach operating on such identified objects.

The Office Actions relied on col. 10, lines 20-30 for the alleged teaching of the steps of identifying objects and performing the operation. These citations clearly do not teach these steps of Claim 1, as alleged. When clarification was requested in the Examiner Interview, the examiner referred to col. 9, lines 61-66 and alleged that a group of actions, or scripts, from a larger set of scripts, are selected for application to attributes of an actor and the actor's surroundings. Such explanation may be what is referred to in the Interview Summary (Paper No. 12) of July 10, 2003 as:

> Perlin teaches an operation identifier, attribute and pattern matching criteria because he discloses modifying an attribute of an object, where the attribute is classified by a group of actions. *Perlin* also teaches that the group of actions is related to an actor....

That explanation is incomprehensible regarding its relation to the required limitations of Claim 1.

To clarify a distinction between the teachings of *Perlin* and Claim 1, it appears that *Perlin* teaches a model, such as BEHAVIOR, that contains a group of scripts. Each script is associated with an action, such as Sleeping, Eating, Talking, Joking, Arguing, Listening, Dancing. Script clauses trigger actions and check, create or modify the actor's properties (col. 9, line 64 through col. 10, line 24). Hence, *Perlin* may teach a group of operations applied to a single object. In contrast, Claim 1 essentially recites **a single operation applied to a group of objects**, which are identified in a specific manner based on information contained in a statement.

Based on the foregoing discussion, it is respectfully submitted that *Perlin* does not anticipate Claim 1 under 35 U.S.C. §102(e) because it does not teach or disclose the limitations of Claim 1.

Claim 12 recites a computer-readable medium carrying instructions which, when executed, cause one or more processors to perform the steps of the method of Claim 1. Therefore, based on the foregoing discussion with respect to Claim 1, *Perlin* does not anticipate Claim 12 under 35 U.S.C. §102(e) because it does not teach or disclose the limitations of Claim 12.

Claims 2-5 and 7 depend directly or indirectly from Claim 1. Claims 13-16 depend directly or indirectly from Claim 12. Hence, for at least the foregoing reasons discussed in reference to Claim 1, Claims 2-5 and Claims 13-16 are also not anticipated by *Perlin*.

Furthermore, since Claims 2, 3, 13 and 14 are asserted as belonging to a different grouping (GROUP 2) and Claim 7 is asserted as belonging to a different grouping (GROUP 3) which do not stand or fall together with GROUP 1, additional rationale for the patentability of these claims is provided.

Claims 2 and 13 recite that the statement includes a first string of characters that contains at least one **wild card character** and that specifies the pattern matching criteria. In the example statement 200, the asterisk "*" is used as a wild card character, such that any object identifier that begins with string "box" satisfies the matching criteria (specification page 13, line 14 through page 14, line 15). Claims 2 and 13 are not claiming the use of an asterisk as a wild card character in and of itself. Rather, Claims 2 and 13 are patentable when combined with the limitations of parent Claims 1 and 12, respectively, which are shown above to be patentable over the references of record.

The Office Actions relied on col. 10, lines 50-60 of *Perlin* for a teaching of the wild card character of Claims 2 and 13. However, this citation makes no mention whatsoever of the use of a wild card character in a string that specifies the pattern matching criteria. Based on the foregoing discussion, it is submitted that *Perlin* does not anticipate Claims 2 and 13 under 35 U.S.C. §102(e) because it does not teach or disclose the limitations of these claims.

Claim 3 depends from Claim 2 and Claim 14 depends from Claim 13. Claims 3 and 14 further recite that the statement includes a second string of characters that includes the attribute

identifier and is in a format that conforms to **object-dot notation**. In the example statement 200, "box*.position" conforms to object-dot notation (specification page 9, lines 6-16). Claims 3 and 14 are not claiming the use of object-dot notation in and of itself. Rather, Claims 3 and 14 are patentable when combined with the limitations of their parent claims, which are shown above to be patentable over the references of record.

The Office Actions relied on col. 11, lines 30-35 of *Perlin* for a teaching of object-dot notation. However, this citation shows "dots" used in conjunction with weights associated with each item in a choice, which are used to affect the probability of each item being chosen. Based on the foregoing discussion, it is submitted that *Perlin* does not anticipate Claims 3 and 14 under 35 U.S.C. §102(e) because it does not teach or disclose the limitations of these claims.

Claim 7 recites that the step of detecting includes detecting that the statement contains pattern matching criteria for a hierarchical identifier. In the example statement 220, a pattern-attribute identifier 228 includes a hierarchy pattern identifier 224, "chicken*/*leg". With hierarchical identifiers, objects that are associated with hierarchically structured identifiers satisfy the matching criteria (specification page 15, lines 10-23). Claim 7 is not claiming the use of hierarchically structured and identified objects in and of itself. Rather, Claim 7 is patentable when combined with the limitations of Claim 1, which is shown above to be patentable over the references of record.

The Office Actions relied on col. 7, lines 15-55 of *Perlin* for a teaching of the pattern matching criteria for a hierarchical identifier of Claim 7. However, this citation appears to describe compositing of layers of motions and/or images, which is in contrast with hierarchically structured or related objects. Based on the foregoing discussion, it is submitted that *Perlin* does not anticipate Claim 7 under 35 U.S.C. §102(e) because it does not teach or disclose the limitations of that claim.

Claim 8 differs from Claim 1 in that the statement detected in Claim 8 includes an **identifier that is associated with a collection of objects** and that executing such statement includes **identifying member objects of the collection of objects specified in the statement**

(specification pages 16 and 17). The Office Actions do not address these limitations, but merely and vaguely address the attribute of a member of a collection of objects. Specifically, it is not shown where in *Perlin* a statement that includes an identifier associated with a collection of objects is taught.

Perlin may describe groupings of actions, such as at col. 7, lines 42-54; and groupings of scripts at col. 9, line 64 through col. 10, line 11. However, *Perlin* fails to teach groupings, or collections, of objects or actors. In addition, and consequently, Perlin fails to teach identifying members of a collection of objects and **performing an operation on an attribute of such identified members of the collection**.

Based on the foregoing discussion, it is respectfully submitted that *Perlin* does not anticipate Claim 8 under 35 U.S.C. §102(e) because it does not teach or disclose the limitations of Claim 8.

Claim 18 recites a computer-readable medium carrying instructions which, when executed, cause one or more processors to perform the steps of the method of Claim 8. Therefore, based on the foregoing discussion with respect to Claim 8, *Perlin* does not anticipate Claim 18 under 35 U.S.C. §102(e) because it does not teach or disclose the limitations of Claim 18.

Claims 9 and 19 depend from Claims 8 and 18, respectively. Hence, for at least the foregoing reasons discussed in reference to Claim 8, Claims 9 and 19 are also not anticipated by *Perlin*.

Furthermore, since Claims 9 and 19 are asserted as belonging to a different grouping (GROUP 5) which does not stand or fall together with GROUP 4, additional rationale for the patentability of these claims is provided.

Claims 9 and 19 recite that the collection of objects is **an array**. In the example statement 240, collection-attribute identifier 248 includes container identifier 244, "boxarray", which is an array that is explicitly declared and used to track graphical elements of a certain type. Thus, any object that is contained in the collection, or container, identified as "boxarray"

is identified and the specified operation is applied to the specified attribute thereof (specification page 16, line 21 through page 17, line 2). Claims 9 and 19 are not claiming the use of an array in and of itself. Rather, Claims 9 and 19 are patentable when combined with the limitations of Claim 8 and Claim 18, respectively, which are shown above to be patentable over the references of record.

The Office Actions relied on col. 7, lines 40-54 of *Perlin* for a teaching of the collection of objects as an array. However, this citation merely shows groupings of actions. No collection of objects and no array are shown. Based on the foregoing discussion, it is submitted that *Perlin* does not anticipate Claims 9 and 19 under 35 U.S.C. §102(e) because it does not teach or disclose the limitations of these claims.

IX. CONCLUSION AND PRAYER FOR RELIEF

Based on the foregoing, it is respectfully submitted that the rejection of Claims 1-5, 7-9, 12-16, 18 and 19 under 35 U.S.C. §102(e) lacks the requisite factual and legal bases. Appellant therefore respectfully requests that the Honorable Board reverse the rejection of Claims 1-5, 7-9, 12-16, 18 and 19.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP

Date: 8/14/03

John D. Henkhaus
Registration No. 42,656

1600 Willow Street
San Jose, California 95125-5106
Tel: (408) 414-1203
Fax: (408) 414-1076

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, Box 1450, Alexandria, VA 22313-1450

on 8/14/03 by Clare C. Tsung

CLAIMS APPENDIX

1. A method of executing an operation on a set of objects, the method comprising the steps of:

detecting that a statement contains

an operation identifier that specifies an operation,

pattern matching criteria, and

an attribute identifier that identifies an attribute; and

executing said statement by

identifying all objects associated with identifiers that satisfy said pattern matching criteria, and

performing said operation on said attribute of each of said objects that satisfy said pattern matching criteria.

2. The method of Claim 1, wherein said statement includes a first string of characters that contains at least one wild card character and that specifies said pattern matching criteria.

3. The method of Claim 2, wherein said first string is part of a second string of characters, wherein said second string of characters includes said attribute identifier and is in a format that conforms to object-dot notation.

4. The method of Claim 1, wherein the step of identifying includes identifying a set of graphical components associated with identifiers that satisfy said pattern matching

criteria, and said step of performing includes performing said operation on said attribute of each graphical component in said set of graphical components.

5. The method of Claim 1, wherein said statement is written in a scripting language and the step of detecting is performed by a script processor.

7. The method of Claim 1, wherein the step of detecting that a statement contains pattern matching criteria includes detecting that the statement contains pattern matching criteria for a hierarchical identifier.

8. A method of executing an operation on collections of objects, the method comprising the steps of:

detecting that a statement contains

an operation identifier that specifies said operation,

an identifier that is associated with a collection of objects, and

an attribute identifier that identifies an attribute of a member object of said collection of objects; and

executing said statement by

identifying member objects of said collection of objects, and

performing said operation on said attribute of each of said identified member objects.

9. The method of Claim 8, wherein said collection of objects is an array.

12. A computer-readable medium carrying one or more sequences of one or more instructions for executing an operation on a set of objects, the one or more sequences of one or more instructions including instructions which, when executed by one or more processors, cause the one or more processors to perform the steps of:

detecting that a statement contains

an operation identifier that specifies an operation,

pattern matching criteria, and

an attribute identifier that identifies an attribute; and

executing said statement by

identifying all objects associated with identifiers that satisfy said pattern matching criteria, and

performing said operation on said attribute of each of said objects that satisfy said pattern matching criteria.

13. The computer-readable medium of Claim 12, wherein said statement includes a first string of characters that contains at least one wild card character and that specifies said pattern matching criteria.

14. The computer-readable medium of Claim 13, wherein said first string is part of a second string of characters, wherein said second string of characters includes said attribute identifier and is in a format that conforms to object-dot notation.

15. The computer-readable medium of Claim 12, wherein the step of identifying includes identifying a set of graphical components associated with identifiers that satisfy said pattern matching criteria, and said step of performing includes performing said operation on said attribute of each graphical component in said set of graphical components.

16. The computer-readable medium of Claim 12, wherein said statement is written in a scripting language and the step of detecting is performed by a script processor.

18. A computer-readable medium carrying one or more sequences of one or more instructions for executing an operation on collections of objects, the one or more sequences of one or more instructions including instructions which, when executed by one or more processors, cause the one or more processors to perform the steps of:

detecting that a statement contains

an operation identifier that specifies said operation,

an identifier that is associated with a collection of objects, and

an attribute identifier that identifies an attribute of a member object of said collection of objects; and

executing said statement by

identifying member objects of said collection of objects, and

performing said operation on said attribute of each of said identified member objects.

19. The computer-readable medium of Claim 18, wherein said collection of objects is an array.